

Relatório Projeto #1 AED 2024-2025

Nome: Pedro David Neves Seco

Nº Estudante: 2021273328

PL (inscrição): 7

- ⇒ Registrar os tempos computacionais das 3 soluções desenvolvidas.
- ⇒ Os tamanhos dos arrays (N) devem permitir obter dados representativos (podem ser, por exemplo: 100000, 200000, 500000, 750000, 1000000). Para os gerar pode: (1) criar um array de tamanho N que vai de 0 a N; (2) remover um qualquer elemento (com exceção do primeiro e último); (3) baralhar o array.
- ⇒ Só deve ser contabilizado o tempo do algoritmo (exclui-se o tempo de leitura/geração do input e de impressão dos resultados).
- ⇒ Sugere-se a realização de várias medições (pelo menos 5) para cada solução e apresentação da média.
- ⇒ Devem apresentar e discutir as regressões para as 3 soluções, incluindo também o coeficiente de determinação/regressão (R^2).
- ⇒ As perguntas de desenvolvimento devem ser **manuscritas**.

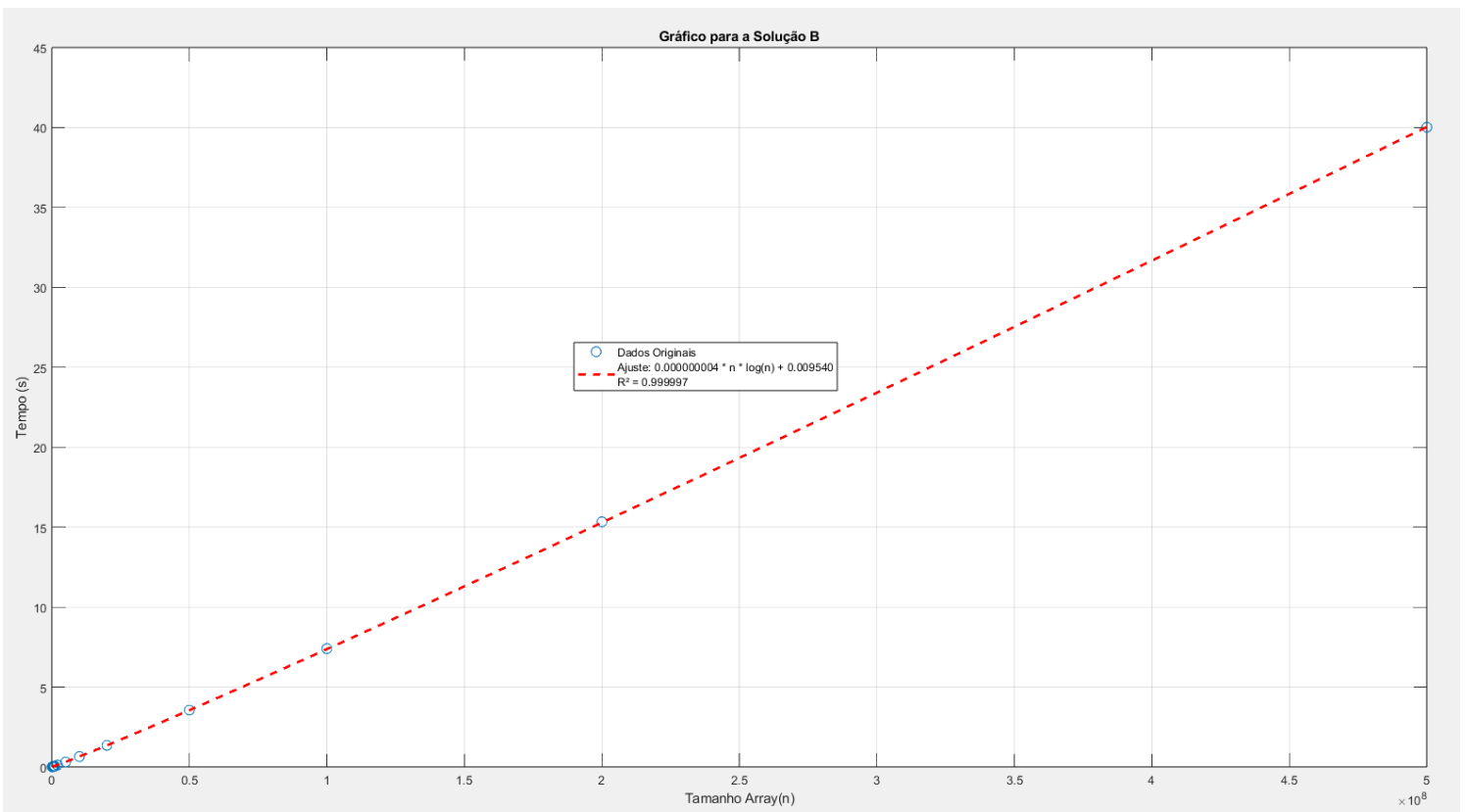
Tabela para as 3 soluções

N	Sol. A (s)	Sol. B (s)	Sol. C (s)
100000	0,394307	0,005356	0,00028
200000	1,87211	0,010111	0,000392
500000	10,325431	0,027369	0,000751
750000	24,415762	0,042821	0,001058
1000000	46,337807	0,057813	0,001351
2000000		0,117814	0,003193
5000000		0,31109	0,006325
10000000		0,64868	0,012615
20000000		1,352843	0,024902
50000000		3,560303	0,0617
100000000		7,404979	0,123378
200000000		15,334739	0,246946
500000000		40,01938	0,61909

Gráfico para a solução A



Gráfico para a solução B

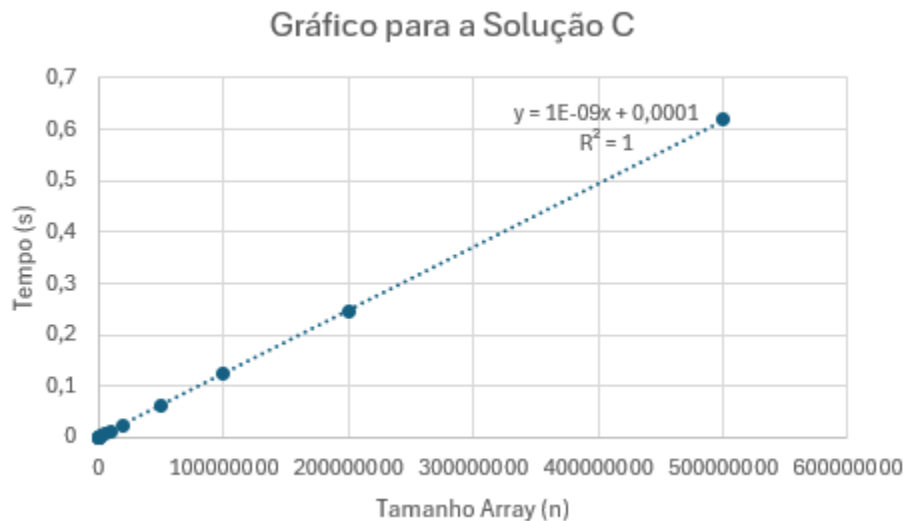


Caso seja de difícil leitura:

$$y = 0.000000004 * n * \log(n) + 0.009540$$

$$R^2 = 0.999997$$

Gráfico para a solução C



Análise dos resultados tendo em conta as regressões obtidas e como estas se comparam com as complexidades teóricas:

A complexidade temporal das soluções A, B e C é respetivamente $O(m^2)$, $O(m \log m)$ (devido ao algoritmo de ordenação) e $O(m)$. As regressões adaptam-se bem aos resultados obtidos para cada algoritmo, uma vez que os coeficientes de regressão são todos muito próximos de 1 (no mínimo 0,9997 para o algoritmo A) e mesmo se a solução C tivesse alguns valores de tempo superiores às outras soluções, a partir de um determinado m , o tempo desta solução seria sempre inferior ao das outras soluções, já que o algoritmo 3 tem a mesma complexidade temporal.

Implementação dos Algoritmos:

- Algoritmo A:

```
public static int findNumber1(int[] arr) {
    int maximo = Arrays.stream(arr).max().getAsInt();
    int minimo = Arrays.stream(arr).min().getAsInt();
    boolean flag = false;
    for (int i = minimo; i <= maximo; i++) {
        for (int j : arr) {
            if (i == j) {
                flag = true;
                break;
            }
        }
        flag = false;
    }
    if (!flag) {
        return i;
    }
}
return -1; // Caso não encontre, retorna -1
}
```

- Algoritmo B:

```
public static int findNumber2(int[] arr) {
    Arrays.sort(arr); // custo nLog(n)
    for (int i = 0; i < arr.length - 1; i++) {
        if (arr[i + 1] - arr[i] > 1) {
            return arr[i] + 1;
        }
    }
    return -1; // Caso não encontre, retorna -1
}
```

- Algoritmo C:

```
public static int findNumber3(int[] arr) {
    int max = Arrays.stream(arr).max().getAsInt();
    int min = Arrays.stream(arr).min().getAsInt();
    long expectedSum = (long) (arr.length + 1) * (max + min) / 2;
    long actualSum = Arrays.stream(arr).sum();
    return (int) (expectedSum - actualSum);
}
```

Notas acerca das características do computador:

- Memória RAM - 16,0 GB
- Processador - AMD Ryzen 5 5500, processador baseado em x64
- Memória SSD – 1TB